**How to work with the API**

**Code Example**

**import requests**

**# Your Azure Computer Vision endpoint URL and subscription key**

**endpoint = 'YOUR\_ENDPOINT\_URL/vision/v3.1/ocr'**

**headers = {**

**'Ocp-Apim-Subscription-Key': 'YOUR\_SUBSCRIPTION\_KEY',**

**'Content-Type': 'application/octet-stream'**

**}**

**# Assuming you have an image stored locally**

**with open('path\_to\_your\_image.jpg', 'rb') as image\_file:**

**image\_data = image\_file.read()**

**response = requests.post(endpoint, headers=headers, data=image\_data)**

**# Check if the request was successful**

**if response.status\_code == 200:**

**ocr\_result = response.json()**

**print(ocr\_result) # This will print the JSON response with the recognized text**

**else:**

**print(f"Error: {response.status\_code}")**

**print(response.text)**

**CREADENTIALS FOR USING OUR REST API ON AZURE**

**SUBSCRIPTION KEYS: “ed81f1ba21fa494ea670787a2718ac13”**

**END POINT:** [**https://shopal.cognitiveservices.azure.com/**](https://shopal.cognitiveservices.azure.com/)

replace path\_to\_your\_image.jpg with the path to your actual image.

1. Parse the Response:
   * The ocr\_result variable in the above example will contain a JSON response with the recognized text and its layout details, such as bounding boxes. You can parse this JSON to extract the exact information you need.
2. Considerations:
   * handle potential errors. The API might return errors due to invalid images, issues with the Azure service, or reaching your quota.
   * If you're integrating this into a web or mobile application, ensure you handle the API requests securely. For instance, avoid embedding the subscription key directly into client-side code. Instead, route requests through a backend server where the key is securely stored.